CLAIMS:

1. A method of providing virtual device services for computerised gaming machines, said method further comprising the steps of:

a. providing a CPU, PSP port, data communications bus and a microprocessor;

b. further providing an internal communications protocol, said protocol comprising a plurality of message transfer frames;

c. abstracting peripheral hardware functions;

d. grouping said abstracted functions into a virtual device type;

e. defining said virtual-device types;

f. defining commands for said virtual device-types;

g. including said commands in said message transfer frames; and

h. monitoring elapsed time between communication of bytes within said frames via said PSP port;

i. monitoring elapsed time between communication of said frames via said PSP port;

j. delaying transmission of said frames if elapsed time is less than a predetermined interframe parameter.

2. The method of claim 1, wherein said data communications bus comprises a multidrop bus.

3. The method of claim 1, wherein said data communications bus comprises an IOCB.

4. The method of claim 1, wherein said command comprises open.

5. The method of claim 1, wherein said command comprises close.

6. The method of claim 1, wherein said command comprises acknowledge.

7. The method of claim 1, wherein said predetermined parameters further comprise a variable level of acknowledgment.

8. The method of claim 1, wherein said predetermined parameters further comprise a variable number of retries.

9. The method of claim 1, wherein said frame further comprises a body segment.

10. The method of claim 9, wherein said body segment further comprises a virtual ID.

11. The method of claim 1, wherein said peripheral device comprises a display.

12. The method of claim 1, wherein said peripheral device comprises a coin

hopper.

13. The method of claim 1, wherein said peripheral device comprises a coin acceptor.

14. The method of claim 1, wherein said peripheral device comprises a bill acceptor.

15. The method of claim 1, wherein said peripheral device comprises a button press.

16. The method of claim 1, wherein said peripheral device comprises a button release.

17. The method of claim 1, wherein said peripheral device comprises an auto-repeat.

18. The method of claim 1, further providing the steps at power up of:

a. identifying set of said peripheral devices performing security functions;

b. enabling said set of said peripheral devices; and

c. disabling said peripheral devices not members of said set.

19. The method of claim 1, further providing the step of including at least one meta-command in said message transport frame.

20. The method of claim 1, further providing the step of including at least one non-common device attribute in said message transport frame.

21. The method of claim 20, wherein said non-common device attribute comprises a hardware type.

22. The method of claim 20, wherein said non-common device attribute comprises a hardware subtype.

23. The method of claim 20, wherein said non-common device attribute comprises a serial number.

24. The method of claim 20, wherein said non-common device attribute comprises a revision level.

25. The method of claim 1, further providing the step of subdividing said message transport frames into subpackets.

26. An interface for communicating with virtual device services in gaming machines, comprising:

a. a CPU and IOCB, each comprising at least one intercharacter timeout counter and interframe timeout counter;

b. a PSP port;

c. data protocol to transfer a plurality of message frames;

d. abstracted peripheral device data within body of said frames.

27. The interface of claim 26, said protocol further comprising:

a. a virtual ID;

b. size variable;

c. sequence number;

d. command field; **[ETX intentionally omitted]**

e. means for measuring elapsed time between characters;

f. means for measuring elapsed time between frames;

g. CRC evaluation means in said message frames;

h. error handling means.